

Chapter 19

SPRINT PLANNING

A release is typically composed of multiple sprints, each of which delivers customer or user value. Every sprint begins with sprint planning, a time when the Scrum team gathers to agree on a sprint goal and determine what it can deliver during the forthcoming sprint. In this chapter I discuss where sprint planning fits into the Scrum framework and how to perform it.

Overview

A product backlog may represent many weeks or months of work, which is much more than can be completed in a single, short sprint. To determine the most important subset of product backlog items to build in the next sprint, the Scrum team performs sprint planning. During sprint planning the Scrum team agrees on a goal for the sprint, and the development team determines the specific product backlog items that are aligned with that goal and that it can realistically deliver by the end of the sprint. To acquire confidence in what it can deliver, the development team creates a plan for how to complete the product backlog items. Together the product backlog items and the plan form the sprint backlog.

Timing

Sprint planning is a recurring, just-in-time activity that takes place at the beginning of each sprint, when we can leverage the best possible information to decide what to work on in the upcoming sprint (see Figure 19.1).

For a two-week to month-long sprint, sprint planning should take no longer than four to eight hours to complete.

Participants

The full Scrum team collaborates during sprint planning. The product owner shares the initial sprint goal, presents the prioritized product backlog, and answers any questions the team might have regarding the product backlog items. The development team works diligently to determine what it can deliver and then makes a realistic commitment at the end of sprint planning. The ScrumMaster, acting as the Scrum team coach, observes the planning activity, asks probing questions, and facilitates to help ensure a successful result. Because the ScrumMaster is not in charge of the

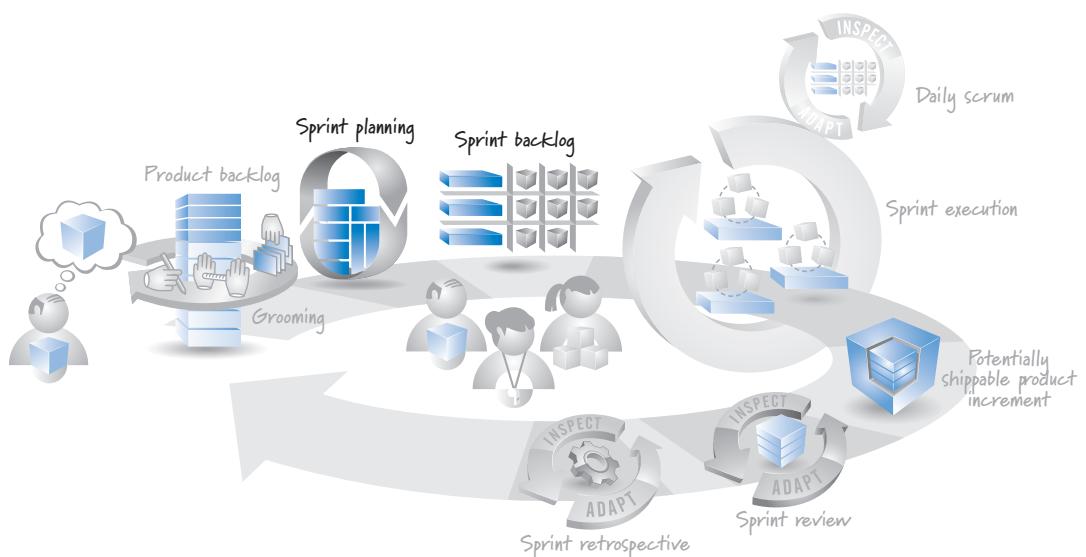


FIGURE 19.1 When sprint planning happens

development team, she cannot decide on behalf of the development team what commitment to make. The ScrumMaster can, however, challenge the team's commitment to ensure that it is realistic and appropriate.

Process

Figure 19.2 illustrates the sprint-planning activity.

Sprint planning relies on a set of inputs that guide the development team in determining what value it can realistically deliver by the end of the sprint. These inputs are described in Table 19.1.

The first and most crucial input to sprint planning is a product backlog that has been groomed prior to sprint planning so that the topmost items meet the Scrum team's definition of ready (see Chapter 6). Typically this means that the topmost items have well-defined acceptance criteria and are appropriately sized, estimated, and prioritized.

Engaged product owners also enter sprint planning having a good idea of what they want the team to deliver by the end of the sprint. They might have a specific set of high-priority product backlog items in mind—"I'd really like to get the top five product backlog items done this sprint"—or they might have a more general notion—"At the end of this sprint I want a typical user to be able to submit a simple keyword query." Knowing the sprint goal helps the team balance competing priorities. A product owner should communicate his initial sprint goal in a way that doesn't unduly influence the development team to commit to more than it realistically can deliver.

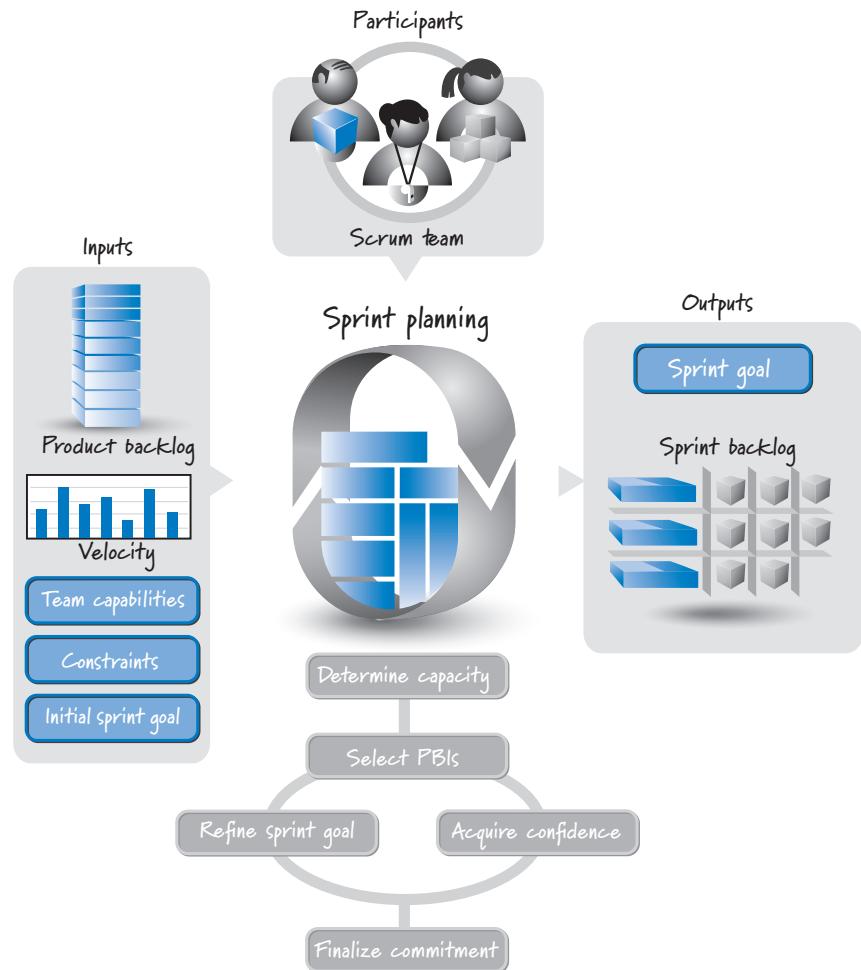


FIGURE 19.2 Sprint-planning activity

TABLE 19.1 Sprint-Planning Inputs

Input	Description
Product backlog	Prior to sprint planning, the topmost product backlog items have been groomed into a <i>ready</i> state.
Team velocity	The team's historical velocity is an indicator of how much work is practical for the team to complete in a sprint.

continues

TABLE 19.1 Sprint-Planning Inputs (*Continued*)

Input	Description
Constraints	Business or technical constraints that could materially affect what the team can deliver are identified.
Team capabilities	Capabilities take into account which people are on the team, what skills each team member has, and how available each person will be in the upcoming sprint.
Initial sprint goal	This is the business goal the product owner would like to see accomplished during the sprint.

The fact that the product owner knows what he wants, however, does not necessarily mean that the development team is capable of delivering it during that sprint. A realistic commitment is achieved only through collaboration (and at times negotiation) between the product owner and the development team members. The sprint-planning participants need to have the opportunity to review and discuss potential value-generating alternatives and decide what is practical given the team's capabilities, predicted velocity, and any known constraints.

To acquire confidence in what it can accomplish, the development team will create a plan for how it will achieve the sprint goal. Collectively the selected product backlog items and the plan form the sprint backlog (shown in Figure 19.2). Most teams break down each targeted product backlog item into a set of estimated tasks, which collectively form the plan. Teams that take this approach typically follow a helpful rule of breaking down tasks so that no one task is more than eight hours of effort, although some might be a bit larger. At this level of granularity the team has a good idea of what really needs to be done and whether it can accomplish those tasks in the time it has available.

At the end of sprint planning the development team's commitment is communicated through a finalized sprint goal and the sprint backlog.

Approaches to Sprint Planning

I will describe two approaches to sprint planning: two-part sprint planning and one-part sprint planning.

Two-Part Sprint Planning

One approach to sprint planning is to separate it into two parts (see Figure 19.3). During part 1 (the "what" part) the development team determines its capacity to complete work and then forecasts the product backlog items that it believes it can

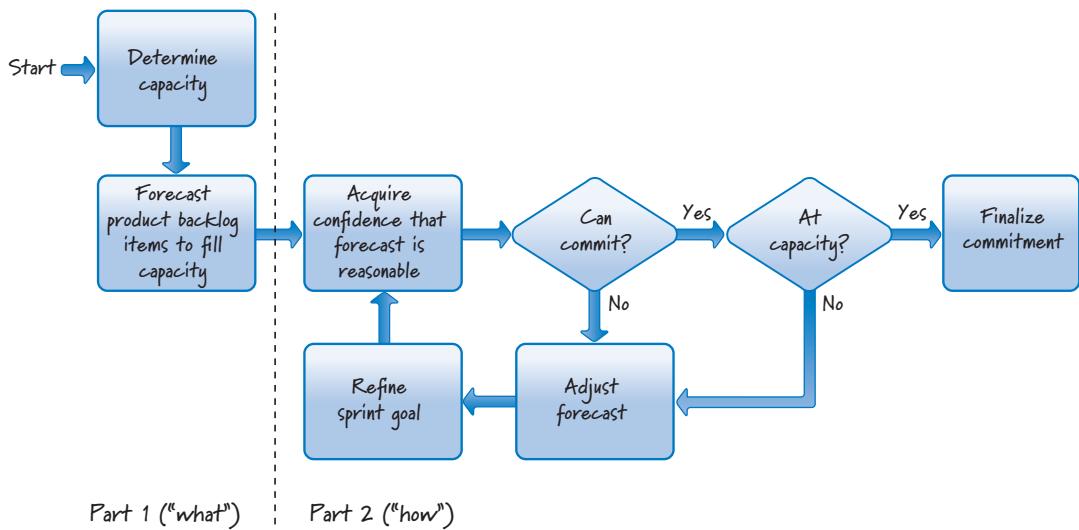


FIGURE 19.3 Two-part sprint-planning approach

deliver by the end of the sprint. So if the team believes it can accomplish 40 story points, it will select about 40 story points' worth of work.

During part 2 (the “how” part) the team acquires confidence in its ability to complete the items that it forecasted in part 1 by creating a plan. Most teams create this plan by breaking the product backlog items into a set of tasks and then estimating (in hours) the effort required to complete each task. The team then compares the estimate of task hours against its capacity, in terms of hours, to see if its initial commitment was realistic.

If the team finds it has selected too much or too little, or has selected items that can't realistically be developed together in the same sprint given one or more constraints, it can adjust its forecast and possibly refine the sprint goal to fit the available capacity and constraints. When the team's forecast is comfortably within its capacity range and constraints, it finalizes its commitment and sprint planning is over.

One-Part Sprint Planning

An alternative approach to sprint planning, and the one I see most frequently, is a one-part approach that interleaves selecting an item and acquiring confidence that it can be delivered. This approach is illustrated in Figure 19.4.

Using this approach, the development team begins by determining its capacity to complete work. Based on available capacity, the sprint goal may need to be refined. Next the team selects a product backlog item and then acquires confidence that the

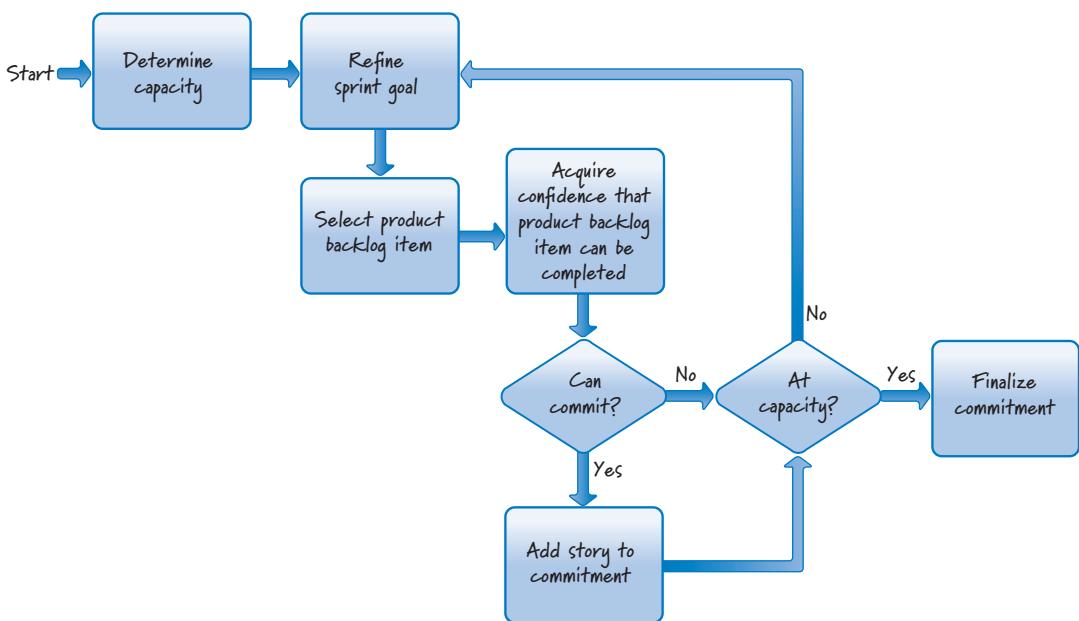


FIGURE 19.4 One-part sprint-planning approach

selected item will reasonably fit within the sprint, given other items already included in the team's evolving commitment. This cycle is then repeated until the team is out of capacity to do any more work. At that point the commitment is finalized and sprint planning is over.

Determining Capacity

An important first activity during sprint planning is determining the available capacity of the team to perform work during the sprint. Knowledge of capacity guides the Scrum team in determining what it can deliver.

What Is Capacity?

Figure 19.5 illustrates the various factors that influence a team's capacity to work on product backlog items during an upcoming sprint. These include the time needed for other Scrum activities, non-sprint-related commitments, personal time off, and the need for a buffer.

Let's say a team is doing a two-week (ten-day) sprint. Right away, we must accept that the team doesn't actually have ten days to dedicate to sprint execution. We know, for instance, that on a two-week sprint about a day of that time needs to be reserved

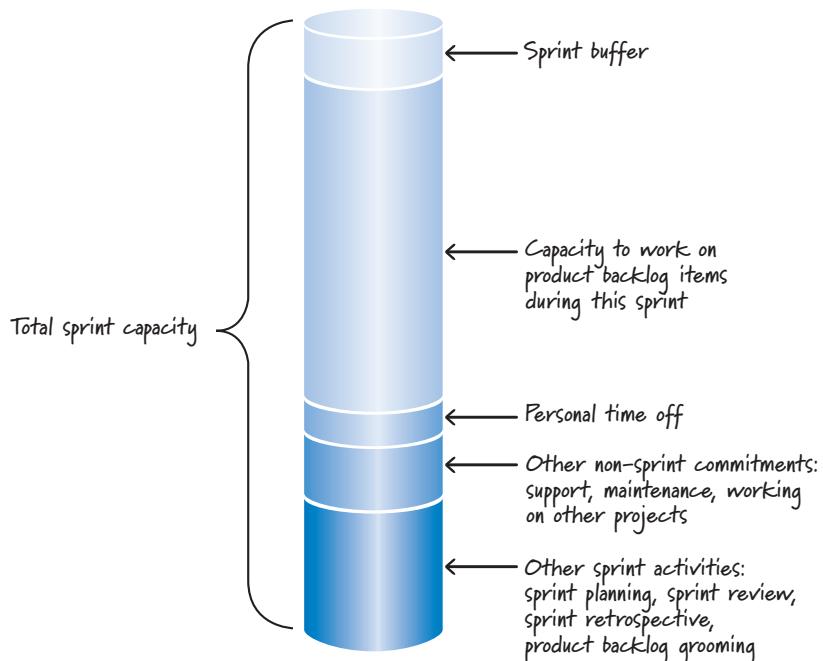


FIGURE 19.5 Development team capacity in a sprint

collectively for sprint-planning, sprint review, and sprint retrospective activities. We also know that the team should reserve up to 10% of its time to assist the product owner with product backlog grooming (writing and refining, estimating, and prioritizing product backlog items) to help ensure that the items are ready.

The team must also determine how much time it should reserve for work outside the sprint, things like supporting the current product, maintaining another product, or other work unrelated to the current sprint. The team should also remember that in an eight-hour day, team members really don't have a full eight hours to work on product backlog items. There is some overhead required to be a good citizen of the organization—attending meetings, responding to emails, interruptions, and so on.

Next, the team needs to know if people have personal time off scheduled during a sprint because that also reduces overall team capacity.

After removing time dedicated to other Scrum activities, work outside the sprint, and personal time off, what remains is the capacity of the team to work on product backlog items for this sprint. However, from this total capacity we should reserve some buffer against things not going quite as planned. For example, any estimating we do won't be perfect, so items might turn out to be a bit larger than we thought. Or something can (and usually does) go wrong. Having a bit of buffer against unexpected problems is wise.

A team can use a number of different approaches to determine a practical buffer size (see Cohn 2006 for some examples). In practice, this buffer can be established empirically after a team performs several sprints and better understands how much buffer should be held in reserve to handle development uncertainty.

Once a buffer is defined, the team can finalize its available capacity for completing work during the sprint.

Capacity in Story Points

What unit of measure should we use for capacity? The two obvious answers are either the same units as the product backlog items (typically story points or ideal days) or the same units as the sprint backlog tasks (effort-hours).

A team's velocity is expressed in terms of product backlog item units (let's say story points). So, if we express capacity in story points, determining capacity is the same as predicting our team's target velocity for the upcoming sprint.

To make this determination, start with the team's long-term average velocity or the team's previous sprint velocity (sometimes referred to as the "yesterday's weather" approach) as an initial estimate of its capacity/velocity for the upcoming sprint. Then consider whether the upcoming sprint might differ from typical or previous sprints (it might not). The result is a reasonable adjusted capacity (predicted velocity) for the upcoming sprint.

For example, let's say our team's average velocity is 40 story points during a two-week sprint. The sprint we are planning, however, occurs during the last two weeks in December in the United States, which means many of our team members will be taking time off for the holidays. We would take on too much work if we used the average velocity of 40; we'd be better off assuming that a velocity closer to 20 (or thereabouts) is a more realistic capacity for the team during this sprint.

Capacity in Effort-Hours

An alternative way to express capacity is in effort-hours. Table 19.2 illustrates how to determine the team's effort-hour capacity to perform task-level work during a two-week or ten-day sprint.

The capacity calculation shown in Table 19.2 is derived as follows. First, the team members express how many days they have available to work on the upcoming sprint (the amount of unavailable time equates to the "personal time off" slice in Figure 19.5). Both Betty and Rajesh are planning to attend a two-day training class, so each of them has only eight days available in the sprint. Simon is planning a three-day weekend so he has nine available days.

Next, the team members determine how much time to reserve for other Scrum activities. They reserve a day of time for sprint planning, sprint review, and sprint retrospective activities. They also deduct the time needed to assist the product owner

TABLE 19.2 Determining Effort-Hour Capacity

Person	Days Available (Less Personal Time)	Days for Other Scrum Activities	Hours per Day	Available Effort-Hours
Jorge	10	2	4–7	32–56
Betty	8	2	5–6	30–36
Rajesh	8	2	4–6	24–36
Simon	9	2	2–3	14–21
Heidi	10	2	5–6	40–48
Total				140–197

with product backlog grooming activities. Together these represent two fewer days available per person to perform task-level work.

The team members then determine how many hours per day they could dedicate to work in this sprint. Each person gives a range that takes into account any overhead work not associated with items in the sprint backlog (the overhead equates to the “other non-sprint commitments” slice in Figure 19.5). For example, Simon is only half-time on this product, so he estimates only two to three hours a day to work on this product during the sprint.

After accounting for personal time off, other Scrum activities, and non-sprint commitments, the team in Table 19.2 estimates a capacity of 140 to 197 effort-hours to work on tasks in the sprint backlog.

I would caution this team against taking on 197 hours of work because it would leave no sprint buffer. A better strategy for this team would be to use a capacity that is probably greater than 140 hours but certainly less than 197 hours when committing to work during this sprint.

Selecting Product Backlog Items

Either approach to sprint planning requires that we select candidate product backlog items for inclusion in the commitment. Selection can be done in several ways. If we have a sprint goal, we would select product backlog items that align with that goal. If there is no formal sprint goal, our default is to select items from the top of the product backlog. We would start with the topmost item and then move to the next item and so forth. If the team were not able to commit to the next-highest-priority item (perhaps there is a skills capacity issue), it would select the next appropriate higher-priority backlog item that looks as if it can be completed within the constraints.

One of my rules when selecting product backlog items for a sprint is that we don't start what we can't finish. So if the next product backlog item is too big to complete in the sprint given the other items that we have already agreed to complete, we should either try to break down the next item into two or more smaller items, each of which would be valuable to our customers, or consider working on another item that we can complete. Also, having a good *definition of ready* will prevent product backlog items from being selected that are poorly defined or have unfulfilled resource or dependency constraints that would prevent our finishing them in a sprint.

The start-only-what-you-can-finish rule is based on the principles that we should limit WIP and that starting something and not finishing it generates a variety of forms of waste. I discussed both of these principles in Chapter 4 when I covered the rule of no goal-altering changes during a sprint. Also, letting incomplete items carry over from one sprint to the next doesn't achieve the goal of having a potentially shippable product increment at the end of each and every sprint.

Acquiring Confidence

One way to acquire confidence is to use predicted velocity to see if the commitment is realistic. If the predicted sprint velocity is 25 story points and our team has selected 45 story points' worth of work, the team should be concerned. At the very least we should start asking questions about why we think the commitment can be achieved. When used this way, velocity provides an excellent check and balance on the proposed commitment.

The risk of using velocity as the sole means of establishing confidence is that even though the numbers look right, the commitment might still be unachievable. For example, if the predicted sprint velocity is 25 story points and the team's commitment totals 21 story points, the commitment would seem reasonable. However, until we dig a little deeper to the task level, we don't really know if the set of product backlog items that total 21 story points can actually be completed—there could be dependency issues, skills capacity issues, as well as a host of other issues that make it impractical for the team to get them all done.

Most Scrum teams gain the necessary level of confidence by breaking the product backlog items down into the tasks that are required to complete them to the Scrum team's agreed-upon definition of done. These tasks can then be estimated (usually in effort-hours) and subtracted from the team's capacity. Breaking product backlog items into tasks is a form of design and just-in-time planning for how to get the items done.

The result is the sprint backlog. Figure 19.6 illustrates an example sprint backlog showing four product backlog items (totaling 21 points) that the team believes it can get done at the end of the sprint. The sprint backlog also shows a plan (in the form of tasks) for delivering the product backlog items to meet the sprint goal.

Is the team making a good commitment or not?

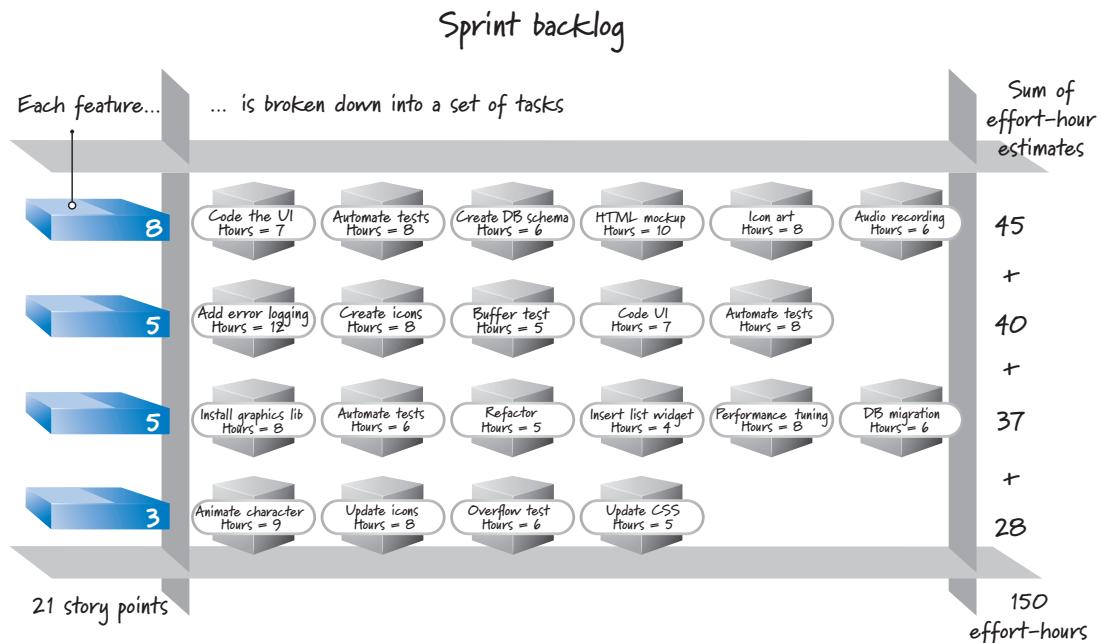


FIGURE 19.6 Sprint backlog showing PBIs and task plan

If the team's predicted velocity is 25 points, a commitment of 21 points seems reasonable. But let's use the task-level details to see if the commitment still looks good. The sum of the tasks for all four product backlog items is 150 effort-hours. Assume that the team for this sprint is the team identified in Table 19.2, which has a task-hour capacity of 140 to 197 effort-hours. A commitment of 150 effort-hours seems comfortably safe—there is likely still a reasonable sprint buffer for handling things that might go wrong.

However, just because the 150 hours are comfortably in the range of 140 to 197 hours doesn't guarantee that the commitment is good. Recall from Table 19.2 that Simon is available only two to three hours a day for nine of the ten days of the sprint. That means Simon has only between 14 and 21 effort-hours available for work on this sprint. What if Simon is the only person who can work on UI tasks? In this case the team might not be able to commit to the four product backlog items in Figure 19.6 because it might run out of "UI capacity" when it runs out of "Simon capacity."

Even though in Scrum we typically don't assign team members to tasks during sprint planning (notice that in Figure 19.6 there are no team member names on the tasks), we need to at least quickly consider our skills capacity or we could make a bad commitment. Just because our commitment is comfortably within the range of estimated aggregate capacity doesn't mean we won't run out of capacity in a particular skills area.

For this reason some teams choose to note on each task who is the person most likely to work on that task. Often this is an unnecessary and potentially wasteful step because unexpected events will occur during the sprint and tasks will need to be reassigned. Also, assigning tasks to individuals might be harmful if doing so reduces team ownership of the tasks. A better strategy (as I will discuss in Chapter 20) is to let team members select the work in an opportunistic, just-in-time fashion during the sprint.

Refine the Sprint Goal

The sprint goal summarizes the business purpose and value of the sprint. The product owner should come to sprint planning with an initial sprint goal. That initial goal, however, can be refined during the course of sprint planning as the sprint-planning participants work together to determine what can realistically be delivered.

Finalize the Commitment

At the completion of sprint planning the development team finalizes its commitment to the business value it will deliver by the end of the sprint. The sprint goal and the selected product backlog items embody that commitment.

As I mentioned in Chapter 2, some people prefer to use the term *forecast* to describe the business value that the development team believes it can produce by the end of the sprint. I prefer and use the term *commitment*. Regardless of which term you prefer, the approaches to sprint planning I describe in this chapter are the same.

The nuanced differences between these terms might affect only the scope of what the development team determines it can deliver and how the Scrum team deals with new information that arrives during sprint execution (see Chapter 4 for a discussion of changes during a sprint).

Closing

In this chapter I expanded the description of sprint planning by discussing when sprint planning takes place and who is involved. I discussed two different approaches to sprint planning. In the first approach the team selects a set of product backlog items and then acquires confidence that it can indeed deliver the full set. The second approach intermingles selecting a product backlog item with acquiring confidence that the item can be added to the incrementally growing commitment. I also explained two different ways to determine a development team's capacity to complete work. In the next chapter I will discuss details of how sprints are executed once they have been planned.